Smart Traffic Lights

by

MacSweeny Dermot, B.Sc. Computer Science

Thesis

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

August 2011

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

MacSweeny Dermot

August 31, 2011

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

MacSweeny Dermot

August 31, 2011

Acknowledgments

Firstly, I would like to thank Kenneth Dawson-Howe, my supervisor for this dissertation for all his help, guidance and time. I would also like to thank my parents and family for all their support throughout the year. My gratitude extends to the entire Ubicom class, I never felt alone in what was an intense and challenging journey. Finally a very special thanks to Aisling whose patience with me throughout this year knew no bounds.

MacSweeny Dermot

University of Dublin, Trinity College

August 2011

Smart Traffic Lights

MacSweeny Dermot,
University of Dublin, Trinity College, 2011

Supervisor: Dawson-Howe Kenneth

Abstract

This project aims to investigate the use of computer vision in traffic signal control. The goal of the system is to analyse video from a camera monitoring an approach to a junction in order to detect and give right of way to approaching traffic. In particular, the system is designed to be of benefit under low density traffic conditions. The traffic lights should change to green when an approaching vehicle or pedestrian is detected if all other approaches are clear. In heavier traffic conditions, the lights could revert to their default pattern.

The system uses a mixture of Gaussians background model to segment the moving objects from the background in a video sequence. These objects are tracked as they move through the scene and classified in order to distinguish between vehicles and pedestrians.

A simulator was created to show how a real junction would benefit from such a system by comparing waiting times with those if the default light sequence were in effect. As well as improving journey times, there are also environmental benefits. Reducing the need for

vehicles to stop and start at the lights will result in a reduction in carbon emissions. The system is also potentially a much more cost effective method of vehicle detection than the use of induction loops, a common alternative.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Many suburban intersections using traffic lights do not have any means of vehicle detection. This results in many drivers and pedestrians being forced to wait at red lights even though it is safe to proceed on their journey. In these circumstances, detecting the presence of vehicles and altering the traffic light sequence to their advantage would reduce waiting times at junctions and have a positive environmental impact.

Some junctions make use of induction loops to detect vehicles and trigger traffic light changes. Induction loops consist of an electrically conducting coil of wire beneath the roads surface. As vehicles pass overhead, an increase of inductance can be detected as the metallic underbody of the car enters the coils magnetic field. Installation of a computer vision detection system would require much less disruption than this method and could potentially be more cost effective.

The stop-start driving patterns which result from traffic lights have a significant impact on the environment. Vehicles travelling at a consistent speed have greater fuel efficiency. The acceleration and deceleration caused by stopping at red lights increases fuel consumption and results in greater levels of carbon emissions dispersed into the atmosphere.

Dynamic traffic light control also has social and safety benefits. Waiting unnecessarily at junctions can cause driver frustration. By eliminating this frustration, the temptation of running a red light when the driver perceives that it is safe to proceed, can be reduced. Similarly the impact on pedestrians could help alleviate jaywalking and provide an overall, safer environment for road users.

1.2 Goals

The main goal of this project is to investigate the use of computer vision in traffic signal control. This involves the development of a proof of concept computer vision application to detect traffic and the creation of a traffic simulator to measure the impact thatsuch a system could have on improving journey times.

The vision system aims to detect moving objects in a scene by analysing video sequences captured from cameras monitoring the approaches to a signalled junction. The system then needs to track the moving objects as they move through the scene and classify them as either vehicles or pedestrians

Modelling the traffic at a signalled junction using a simulation can measure the improvement in journey times under different traffic conditions. The simulation is required to measure the average time it takes for vehicles to clear a junction. These times can be calculated for both the default traffic light sequence and a sequence triggered by the vehicle detection system. By comparing the times for the two different sequencing operations, the impact which the detection system has on journey times can be measured for varying levels of traffic flow.

1.3 Approach

The detection system was developed in C++ using the OpenCV2.1 computer vision libraries which provide an array of computer vision functions. The system was developed and tested around video samples from the CLEAR evaluation and workshop [1]. Figure 1.1

shows an image from one of the sample videos of an oncoming car and an approaching pedestrian is shown in Figure 1.2.

Figure 1.1: Approaching Car

There were three main steps involved in developing this system. First it is necessary to separate the moving objects in the scene from the background. Once these moving objects are found they must be tracked from frame to frame. Finally, each tracked objects is classified as either a vehicle or a pedestrian.

1.3.1 Segmentation

Separating the moving objects from the background in a sequence of video frames involves determining whether each pixel belongs to the foreground or the background. In this project a probabilistic approach is taken to this problem and a mixture of Gaussians model is used to classify each pixel. The history of each pixels intensity values are then modelled as a number of Gaussian distributions. The distributions with the greatest weights above a certain threshold are used to represent the background. A pixel from

Figure 1.2: Approaching Pedestrian

a newly captured frame can be classified as a background pixel if it matches one of the background distributions. The model is then updated with the newly classified values and a binary mask is generated for each frame. The mask consists of white pixels for those labelled as foreground and black for the background. Noise in the mask image is removed by using erosion and dilation.

Shadows of moving objects can often be classified as part of the foreground. Removal of shadow pixels is important in order to facilitate better tracking of objects and improved classification. Identifying shadow pixels can be more easily done in the HLS colour space. By examining the differences in luminance and saturation values between pixels in the current frame and those in a background image, it can be determined which pixels are caused by shadows in the scene. These pixels can then be removed from the foreground mask.

Background pixels are then grouped to find the connected regions that represent objects in the image. This is achieved by finding the all the contours in the foreground mask image. The contours are the outline curves of the regions in the image representing the moving objects we wish to detect.

1.3.2 Tracking

Tracking is then performed by matching each contour to an existing object track or assigning a new track if no suitable track already exists. The position of the centroid of each contour is calculated. A contour is then allocated to a track if the distance between its centroid position and the previous position of the centroid stored in the track from a previous frame is less then a predefined minimum distance. The centroid value stored in the track is then updated. Non-persistent tracks, ones which have not been updated for a set number of frames are disregarded.

1.3.3 Classification

Persistent tracks, those that have been updated frequently within a certain window are classified in order to distinguish tracks of moving vehicles from those of pedestrians. The classification method which was successfully implemented into the system is based on the height to width ratio of the tracked contours. Tracks of contours which have a ratio greater than one are deemed to be pedestrians and those which have a ratio less than one are labelled pedestrians.

1.3.4 Dissertation Structure

The structure of this dissertation is outlined as follows:

- Chapter 2 reviews the background research and state of the art relative to this dissertation.
- Chapter 3-5 concern the three main steps involved in the detection system:
  - Segmentation
  - Tracking

    - Classification

- Chapter 6 describes the design, models used and results of the traffic simulation program.

- Chapter 7 concludes the dissertation.

Chapter 2

Background and Related Work

2.1 Introduction

The following chapter reviews the state of the art literature concerning the utilisation of computer vision to analyse road traffic scenes. There is a vast amount work on detecting and classifying vehicles and pedestrians in traffic scenes for such purposes as measuring traffic levels and flow and measuring vehicle speeds. Section 1.2 will look at the use of omnidirectional cameras in computer vision. The literature discusses the geometric transformations necessary to map the captured video frames and how the cameras have been deployed for vehicle detection and traffic control. Section 1.3 reviews the state of the art in motion detection in traffic scenes. The section addresses such concerns as choosing a background model for background subtraction, handling varied lighting conditions as well as shadow and ghost removal. In order to perform beneficial traffic light control, it is necessary to classify moving objects that are detected as pedestrians or vehicles. Techniques used for classification of vehicles and pedestrians are discussed in the works in section 1.4.

2.2 Omnidirectional Cameras

Omnidirectional cameras can be used in computer vision to provide an extended field of view of a scene. Their use can eliminate the need for multiple cameras.

Ghorayeb et al. [5] present an omnidirectional camera to be used for traffic monitoring at a crossroads. The camera is positioned in the very centre of the junction and uses a custom made mirror which maximises the number of pixels in the image that represent the approaching roads. The mirror showed an improvement in the detection rates when compared to spherical, parabolic and hyperbolic mirrors but the system may have limited flexibility. Deployment at junctions where the approaches are not all perpendicular may prove difficult as roads might lie outside the field of view. Vehicle detection is carried out without geometric transformation and there was no consideration given to the classification of objects.

Khoshabeh et al. [8] describe a system using a combination of an omnidirectional camera and a pan tilt zoom camera to analyze vehicles and traffic flow. Coarse classifications are carried out by the omnidirectional camera and more refined classifications by the PTZ camera, thus eliminating the trade-off between resolution and field of view. A mixture of Gaussians background model is used to segment the images. Blobs are tracked using a Kalman filter. Foreground objects can be classified as one of a number of predefined classes such as person, car, crowd, bus and no label. Tracks can be classified by the size of a bounding box area. The relationship between object position and size in the image from the omnidirectional camera is not uniform. It is necessary to plot track sizes as a function of image position.

In a similar design, again an omnidirectional sensor is combined with a pan tilt zoom (PTZ) camera in work carried out by Scotti et al. [17]. The catadioptric sensor chosen utilises a mirror with a double parabolic envelope and a conventional camera pointed vertically towards the mirror. The paper describes three phases of camera calibration, intrinsic, extrinsic and joint calibration. The first phase involves rectification of the polar image. Re-sampling is used to fix the inconsistent resolution in the resulting polar image. Extrinsic calibration maps the image plane to points in the real world and involves the calculation of the tilt angle as a function of the distance from the image centre. The final phase describes a method for aligning coordinates in the rectified image with those captured by the PTZ camera. An operator of the system can then select points in the low resolution rectified image from the omnidirectional sensor and track them with the higher resolution PTZ camera. Background subtraction and temporal correspondences between blobs are used to track moving objects. The results show a frequency of object detection greater than 95%, frequency of object loss of 5% and a false positive rate less than 2%. The system proposed is aimed towards surveillance applications and offers a more cost effective solution as it would take multiple traditional cameras to cover the same field of view.

Scotti et al. [16] also created a system which uses a combination of an omnidirectional camera and a pan-tilt camera for classification of moving objects, in particular tracking pedestrians in the context of car park surveillance. Classification is simplified by taking advantage of geometric properties of the polar image captured by the omnidirectional sensor and the those of the pedestrians. Pedestrians in the ground-plane are modelled as cylinders which are approximated as ellipses when projected to the polar image plane. Cars are also modelled as ellipses but with different orientation. The orientation angles of the ellipses can then be used as the main classification feature because the projection of the ellipses is always oriented along the radius of polar image. This works quite well because of the context of the system but in rare cases in which a vehicle is moving towards the centre of the image, object size is used as a classifier. There is some ambiguity in how the system was tested. The authors specify results carried out over 200 tests but it is unclear if this figure refers to the number of video sequences or the number of objects that they were attempting to track. It is also unclear if the 5% frequency of object loss and 2% false classification is referring to missed objects per frame or per sequence.

2.3 Motion Detection and Segmentation

Detection of moving objects in a video stream can be achieved by first determining which pixels in the scene belong to moving objects and then by grouping the pixels into objects of interest. Background subtraction is the most common method of determining which pixels belong to moving objects in a video sequence. The technique involves calculating a binary difference image by subtracting a background image from the current frame of the video sequence. In its simplest form a single static background image that can contains no moving objects can be used. More robust background models have been developed to dynamically change the background image to adapt to lighting changes and objects entering and leaving the scene. However there are various techniques used for generating the background models.

A system is proposed by Lipton et al. [11] for tracking moving objects in video streams and classifying them as human or vehicle. The approach uses temporal differencing to segment the moving objects from the background. Temporal differencing simply involves subtracting calculating an absolute difference image by subtracting the pixel intensity values for each pixel over consecutive frames. Binary thresholding on the difference image is then used to determine which pixels are motion pixels in the current frame. Then motion regions are determined using connected components analysis.

Liu et al. [12] focus on developing a robust solution for vehicle detection that can handle varied environmental conditions such as lighting, sun position, shadows, weather conditions and surface reflections. A background subtraction technique is used based on RGB colour images. A temporal difference image is computed by summing the absolute difference in intensities between the background image and the current frame in each colour channel. A threshold value is dynamically estimated from the histogram of the difference image. The threshold is then used to create a binary mask. This technique is used because the author suggests that colour image segmentation is less sensitive to illumination changes than using grayscale images. Ghosts are removed by comparing values of moving object pixels from frame to frame. Ghost pixels that should be removed from the binary mask have very small or no differences in intensity from frame to frame. A temporary background is computed combining the previous background pixels from the moving areas and the current image pixels for static areas. The current background image can then be computed as a weighted average of the temporary background and the current background. Dual mode processing is then used for more refined detection, under low contrast conditions, vehicles are mapped as pairs of circular regions representing headlights and under higher contrast conditions vehicles are mapped as rectangular patches. The mode is switched based on HSV values. The authors suggested the applications use for traffic control; and proposed using a different camera for each intersection approach.

A vehicle detection and classification system is proposed by Messelodi et al. [14] aimed at collecting traffic data for statistical purposes at urban intersections. Object detection is achieved by using a background subtraction technique. An initial background is generated using a temporal median technique over an initial 20 second video sequence. A novel extension to a Kalman filter is used for updating the background model. Normally background models based on Kalman filters can only accommodate slow variations in lighting changes. The paper assumes that the luminance levels of pixels are proportional to the ambient light, a global illumination change causes a variation in pixel values proportional to the proceeding values. By computing the ratio of pixel values from one frame to the next for each pixel and analysing the distribution of the ratios, moving objects can be distinguished from sudden lighting changes such as the switching on of artificial lights or the sun disappearing behind clouds. A two pass convulsion using a 1D Gaussian derivative filter is used to compute the gradient of the the absolute difference map between current frame and the background. The gradient map is thresholded and noise is removed by morphological operations.

The method proposed for background modeling by Stauffer et al. [19] models the values of each pixel individually as a mixture of Gaussians. Certain Gaussians are then determined to be part of the background based on their persistence and variance. Pixels are then classified as background if the Gaussian distribution with the best representation of the pixel is determined as representing the background. Pixels which are found to be represented by one of the Gaussian models are used to update the model. New pixels are classified as part of the background if the pixel value is within 2.5 times the standard deviations of the Gaussian model. The method can incorporate different lighting conditions amongst different regions in a particular scene as it works on a per pixel level. The best matching Gaussian distribution for each pixel in the current frame is updated with the current pixel value. For pixel values which do not fit any of the Gaussian models, a new distribution with the given value as the mean replaces the least probable distribution for that pixel. The models are then adapted to only consider the most recent history of a pixel and to give increasingly greater weight to values newer values . In this way, a robust and adaptive model is produced which the authors propose can handle both temporal and regional lighting changes in a video sequence.

The background model proposed by Magee [13] extends the mixture of Gaussians method [19]. Instead of using a mixture of Gaussians model, a mixture of axes orientated cylindrical distributions is used. It is claimed that this technique is more sensitive to lighting changes, is less expensive and can handle camera jitter which can lead to edge pixels being misclassified when using the mixture of Gaussians distribution. Instead of using connected components, individual pixels are associated with object models. Pixel are classified as belonging to a particular model based on position, size and colour distribution. Position is represented by the centroid of the object as projected onto the ground plane. Object size is represented by 1D Gaussians describing the variation of contributing pixel locations about the object centroid in, and perpendicular to, the direction of travel. Colour distribution is modelled by a Gaussian mixture used to represent colour over the entire object.

Gupte et al. [6] perform vehicle detection for the purposes of vehicle counting and classification. The background model is updated by taking a weighted average of the current and instantaneous backgrounds. An object mask is computed by subtracting the current background from the current video frame. The instantaneous background image is then defined as the pixels with a zero value in the object mask. A dynamic threshold is used on the difference image obtained from background subtraction. The threshold is set by locating peaks in the histogram of the difference image and searching for values on the histogram that occur after the peak that are 10% of peak value.

Segmentation is performed by K means clustering on each image and a model based approach is used for recognition by Setchell et al. [18]. During segmentation, the K value is set high enough that the image may be over-segmented but region merging is used in order to map regions to model features. Regions are recursively merged with their neighbours and matched against model features. This results in each model feature having a list well matched merged regions associated with it. An interpretation tree is then used to find the optimal one to one mapping between regions and model features. Each node in the tree represents a mapping of a model feature to a region feature. Due to the complexity of searching all possible mappings, it is necessary to prune the tree by removing branches where poorly matched features are discovered. When tested on a real traffic scene, the method showed an 84% success rate in vehicle detection.

2.4 Object Tracking

Gupte et al. [6] address some of the difficulties involved in region tracking from frame to frame. These include new regions appearing, existing regions disappearing, splitting and merging. Associations between regions in consecutive frames are modelled by a bipartite graph are used to overcome these difficulties. The edges matching the previous frame regions to the current frame regions are weighted by the area of overlap of the two regions. Thus regions can be tracked from frame to frame by finding the maximal weight graph. The system provided coarse classification separating cars from larger vehicles such as SUVs and trucks. The classification was performed using the dimensions of the bounding rectangles of the tracked regions. The system was tested on a 20 minute video sequence and processed at 15 fps. A 90% detection rate and a 70% classification rate was achieved.

Messelodi et al. [14], maintain a list of detected objects described as polygons of their convex hulls. Each object in the list is then compared against a list of active objects, if there is no overlap with an active object then it is marked as a new object. A threshold map that considers camera perspective is used to remove objects that are too small to be of interest. The region based tracking is relatively expensive and is not performed on every frame. A feature based method is also used to provide less accurate but faster tracking.

Beymur et al. [2] describe a vehicle tracking system to be used for measuring traffic statistics and for traffic signal control. Rather than tracking the entire vehicle, a set of sub-features are tracked which provides more robust tracking when dealing with partial occlusions. Corners are chosen as the most reliable features to track from frame to frame. A Kalman filter is used to predict the location of a given feature in the next frame of the sequence based on data gathered from previous frames. The tracked features of an individual vehicle are grouped together based on proximity and common motion in order to distinguish the vehicle from others in the scene. Features must have an almost identical motion as they are tracked through a defined region of the scene. The system was evaluated on both sample videos and in real-time. The offline testing evaluated the systems performance by comparing the tracked feature groups with the ground truth. On all but one of the video sequences the system had a match of between 85% and 95%. The online tracker showed a high level of accuracy in measuring vehicle velocity but was less accurate at measuring traffic density and flow when compared with data gathered concurrently from an inductive loop.

2.5 Vehicle and Pedestrian Classification

A tracking approach using a Kalman filter is presented by Veeraraghavan et al. [21] for tracking vehicles and pedestrians at intersections for the purpose of incident detection.

Input is gray scale images from stationary camera. Segmentation is performed using a mixture of Gaussian models method. Connected components extraction is used to obtain regions with motion referred to as blobs. The various attributes of the blob such as centroid, area, elongation, and first and second-order moments are computed during the connected component extraction. Blobs are approximated by rotating bounding boxes. Blobs are tracked from frame to frame based on their proximity. Position estimation of the moving objects is done using an extended Kalman filter. The extended filter is necessary because of the non-linearity in the mapping from the real world coordinates to the image coordinates. Occlusion detection is performed by using a standard discrete Kalman filter for shape estimation. The paper focuses on the tracking of pedestrians and vehicles and handling occlusion but does not consider classification.

Moving objects in a scene are classified as either pedestrians or vehicles using boosting based detection in Javed et al. [7]. Global features are used on regions of interest obtained from background subtraction instead of local Haar like features for object representation. Principle component analysis is used to obtain the global features. Feature vectors are calculated by projecting each training example into both a pedestrian and a vehicle subspace. The subspaces are generated from principle component analysis on the gradient magnitudes of training images. Bayesian classification is then used to calculate the most likely class given a particular feature vector. The adaboost algorithm is used for learning the boosted classifier. In the training phase of the algorithm, an initial distribution of weights is generated over the training set, then a base classifier is selected that gives the least error. The weights associated with data that was misclassified by the first base classifier are increased. Since the error is proportional to the weights of the misclassified data, the classifier chosen by the algorithm for the next iteration should perform better on the misclassified data on the next iteration. If a base classifier selected by the boosting algorithm predicts the label above a certain confidence threshold, then the unlabeled data can be added to the training data and used to update the base classifiers. In this way, online training provides classifiers that can adapt to a specific scenario.

Buch et al.[4] perform vehicle classification on a per frame basis. A classifier generates a hypothesis of a vehicle or pedestrian being present in the scene by placing each 3D model onto the scenes ground plane and projecting it to the camera view. Every model is placed on a grid of positions on the ground plane to produce a match measure for every silhouette. The highest measure indicates the most likely position of the vehicle given the silhouette and the highest match measures of different classes are compared to make a decision about the class of a silhouette. The system does not consider occlusion and assumes constant orientation of vehicles. The system was evaluated on video from i-LIDS datasets which is licensed by the uk home office. The dataset also provided the ground truth consisting of bounding boxes and class labels for vehicles.

Koller et al. [9] proposes a system for traffic analysis utilising computer vision and a dynamic belief network. The goal is to derive a vehicles behaviour, such as lane changes from observations. Motion segmentation is done by background subtraction with a background model based on a kalman filter to adapt for lighting changes. Closed contour extraction is used to identify individual vehicles from blobs. Contour extraction is based upon motion and gray-value boundaries, which are obtained by thresholding the spatial image gradients and the time derivatives of the image. Thresholded blobs are enclosed by convex polygons which are approximated by using closed cubic spline with 12 control points. A Kalman filter is again used to obtain the control points.

The system by Messelodi et al. [14] as mentioned earlier, uses 3D model based classification. The models are compared with the convex hulls of the detected moving objects for all locations and orientations. The projection of each model throughout the scene is pre-computed as this process is too expensive for real time processing. The system was evaluated on three sequences of video obtained from real intersections. The ground truth was obtained by visual inspection with a reliability of 95%. Errors were mainly due to partial occlusions, shadows and presence of pedestrian groups.

Bhuvaneshwar et al. [3] describe a fixed camera system, developed to consider pedestrian traffic at intersections with the goal of counting pedestrians in order to aid traffic control. A background model similar to that used by Stauffer is used and the difference images are used to generate binary images by using median filtering and thresholding. Blobs are formed by connected components analysis. The blobs are labelled as vehicles or pedestrians based on two features, height and area. Labelled objects are then tracked from frame to frame using a maximum likelihood-matching algorithm. The paper emphasises the importance of shadow removal when labelling objects by area as the presence of shadows can cause objects to be misclassified. A novel solution to this problem is proposed involving counting the number of object pixels in each column and removing the pixels in columns that are below a certain threshold.

A method of object detection without the use of tracking or motion is presented by Papageorgiou et al. [15]. Pattern matching is performed on each frame without using information from previous frames. A training set of 1800 scaled and aligned, colour images is used. Haar wavelets are used rather than a pixel or edge information in order to represent the object classes. For each pattern, the wavelets are calculated for diagonal, horizontal and vertical orientations and at multiple scales. An over-complete approach is used whereby each haar wavelet is shifted by a quarter of the length of the supporting mask rather than the standard approach which does not allow the wavelets overlap. This generates a feature vector with 1326 dimensions for each pattern in the training set. A 128 x 64 pixel detection window is shifted over an iteratively scaled image in order to detect pedestrians. This is far too complex to compute in real time and does not take advantage of any dynamic information from a sequence of video frames. The authors describe methods to reduce the complexity of the computation and how the scaled down version was integrated into a real vehicle as part of a driver assistance system. The system was modified to consider a reduced feature set consisting of 29 wavelets and results were compared on both grey level and colour images. ROC curves show the trade-off in performance necessary to increase the speed of the system. The curves graph the number of correctly detected objects against the quantity of false positives detected.

The system presented by Kumar et al. [10], aims to track vehicle behaviour from traffic videos. In particular two types of behaviour are focused on, interactions between two or more vehicles and interactions between vehicles and stationary objects in the scene. A background model using a single Gaussian distribution is used in order to segment moving objects from the background. The segmentation is carried out in YCbCr colour space which the authors claim is optimal for shadow detection. Shadows can be separated from objects of interest by recognising that they may have a high variance in the luminance channel but will have low variance in the two chromacity channels. The approach used for classification is based on a Bayesian Network. The network consists of a number of nodes representing target features and root node representing the class of target. This allows for the probabilistic classification of targets based upon size, shape, position and velocity. The centroid of an object represents its position and its size is given by an ellipse approximating its convex hull. Velocity is estimated by a Kalman filter which tracks the objects centroid. Supervised learning is used to train the classifier. Further it is described how the behaviour of moving objects can be classified based on their interaction with static objects in the scene by providing contextual information about the scene and using the features of the tracked object. The behavioural classifications are motivated by a desire to detect dangerous vehicle behaviour and behaviour at a security checkpoint. Tests were performed on a number of traffic video sequences with over 1000 targets. The classification results showed correct classification rates varying from 86.7% for pedestrians to 96.3% for cars.

Chapter 3

Segmentation

This chapter describes in detail how moving objects are separated from the background in scenes from the input video. This process involves a number of steps. First, a mixture of Gaussians background model is used to determine which pixels in the input image are belonging to the background in order to create a binary mask of the foreground image. Then noise is removed from the binary mask. Shadow pixels are also classified as part of the foreground and must too be removed. Finally, the contours of the connected foreground regions in the mask are found. Figure 3.1. shows the pipeline of the process of segmenting a frame.

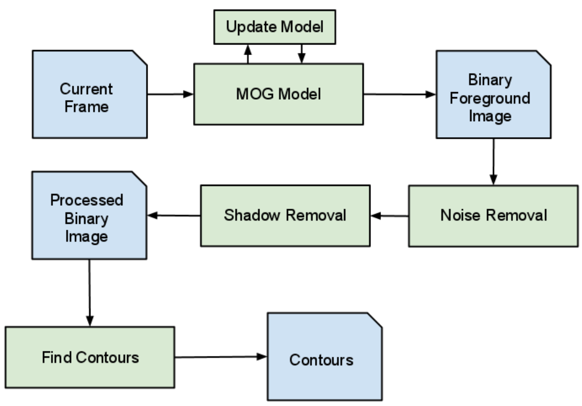

Figure 3.1: Segmentation Process Pipeline

3.1 Mixture of Gaussians Background Model

A Gaussian Mixture Model is used to determine which pixels belong to the background and which belong to moving objects. This approach is particularly useful when a background pixel can have more then one value. This allows the model to handle changes in lighting conditions, camera jitter and repetitive motion such as they the swaying of a tree branch. The model also performs well when confronted with slow moving objects or objects entering or leaving the scene. The model builds up a recent history of the intensity values for each pixels over time. The most frequent values for each pixel are assumed to be the values which best represent the background. These are the values which represent the largest portion of a particular pixels history. An example can be seen in Figure 3.2 which contains a histogram showing the frequency of intensity values for a given pixel. As in this case where the data is multi-modal, a single normal distribution does not always provide an accurate approximation.

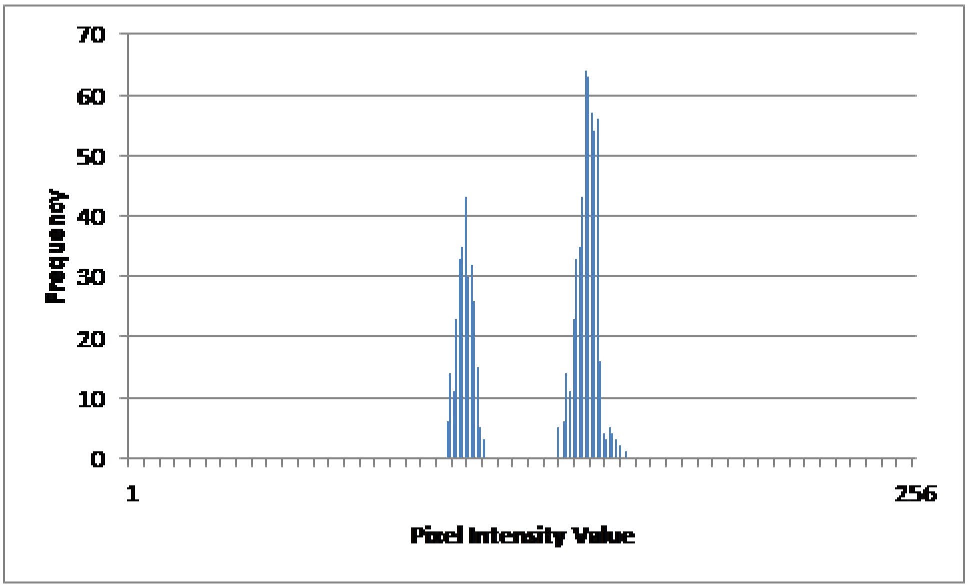

Figure 3.2: Histogram of Intensity Values for a given Pixel

The mixture model as depicted in Figure 3.3, uses multiple distributions. A pixel from a new frame can be categorised as belonging to the background if its value is within 2.5 times the standard deviation of any of the background distributions. If its value is outside this range, it belongs to a moving object.

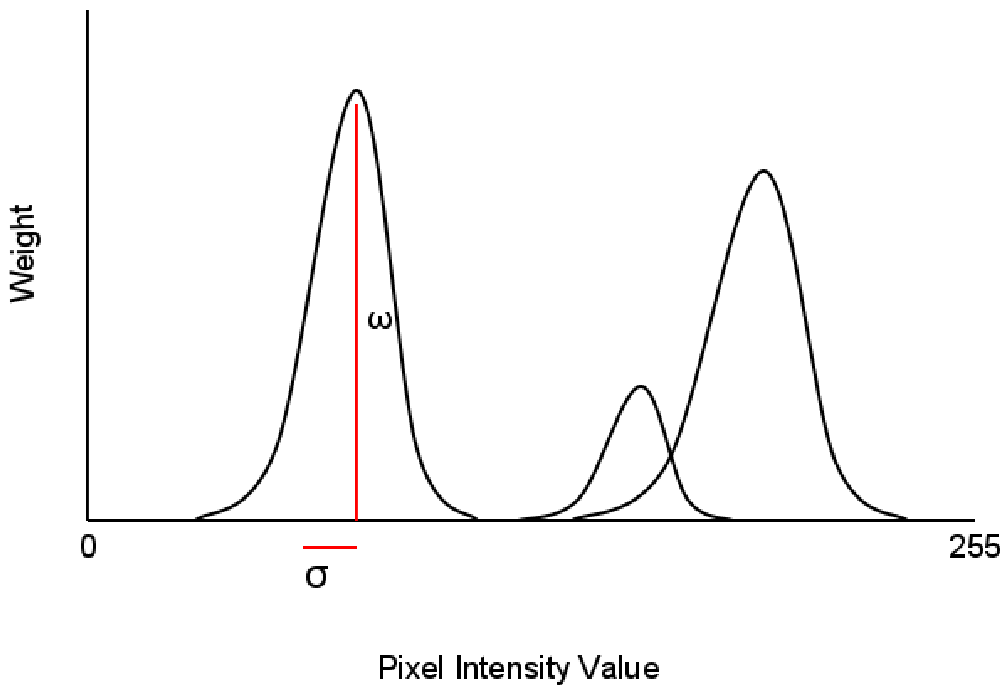

Figure 3.3: Mixture of Gaussian Distributions

Weights are assigned to each distribution to allow more recent values to have a greater contribution. The model is updated with each new frame and all weights and distributions are recalculated. The weight of the kth distribution at time t is updated by formula in Equation (3.1) where α is the learning rate and M_k is 1 if the distribution matched the pixel value and 0 for all other distributions.

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t} + \alpha(M_{k,t}). \tag{3.1}$$

Three distributions were chosen to represent the background in the model used in this project. The number of distributions used is typically between three and five and the choice is motivated by the performance requirements and the content of the video to be processed. The model only considers the most recent pixel values and a window-size of 100 frames was used in this application. However, experimenting with a reduced window size showed improvement in processing speed but produced a less accurate foreground mask.

Overall the mixture of Gaussians Model performed very well in segmenting the frames. The only downside was the algorithms computational cost which had a negative effect on the speed at which frames could be processed. Improvements in efficiency would be necessary in order to create a viable traffic detection system as real-time performance would be essential.

Figure 3.4 shows a sample frame from the input video and Figure 3.5 shows the binary mask of the segmented regions.

Figure 3.4: Input Frame

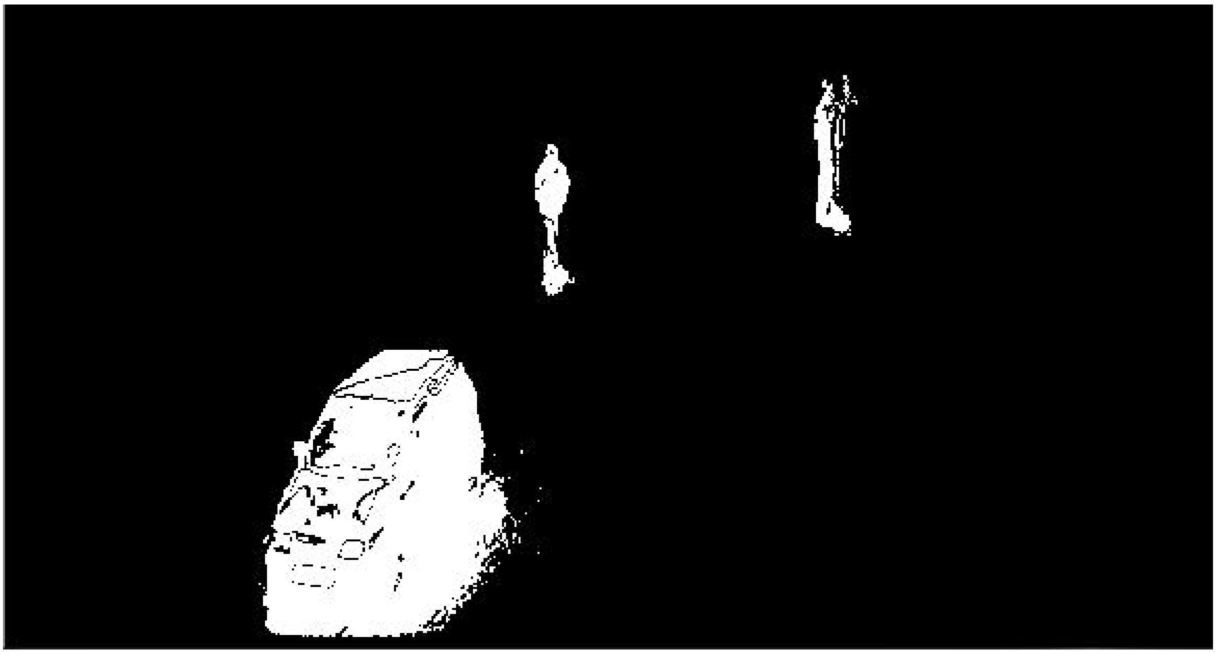

Figure 3.5: Binary Foreground Mask

3.2 Noise Removal

Although the Gaussian model proved very successful in segmenting the image there were occasionally small amounts of noise in the binary foreground mask. There were some isolated and small clusters of misclassified pixels. These were mostly located in the regions close to where foreground pixels were correctly classified. Opening operations were used to remove these undesired pixels from the mask. This was followed by Closing which is used to fill in small gaps in the image. It is important to have a consistent outline of each object at least as the contours of each connected group of object pixels will need to found for the tracking process. Opening uses two morphological operators, erosion followed by dilation. The same two operators are used for Closing but in the reverse order.

During an Erosion operation object pixels directly next to any background pixels are removed. Dilation is the opposite of erosion, where foreground objects are expanded to include any directly adjacent background pixels. These operations are performed by passing a structural element over each foreground pixel in the image. During erosion, foreground pixels are removed unless all of there neighbours as indicated by the structural element are also belonging to the foreground. Dilation adds all neighbouring pixels to the foreground if they are directly adjacent to background pixels. Figure 3.6 demonstrates the results of each process on a small set of pixels for a given structural element.

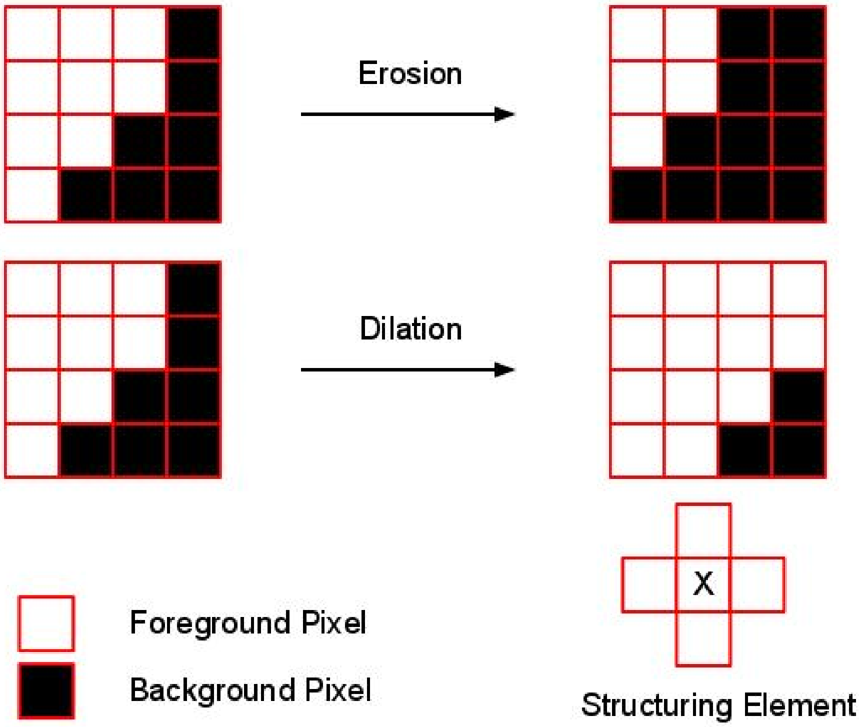

Figure 3.6: Dilation and Erosion Operations

Figure 3.7 shows an image of the binary mask containing some noise and small gaps in the foreground region. Figure 3.8 shows an image of the same frame after Opening and Closing operations. As is evident from these images, it is very difficult to ensure perfect segmentation of every frame. Fragmentation, holes in foreground regions and noise can be minimised but not entirely eliminated. It can also be seen that the shadow of the moving vehicle is being classified as part of the foreground. For the purposes of this project, it was helpful to remove shadow pixels from the foreground mask as this will make tracking and classification more difficult.

Figure 3.7: A noisy binary mask image.

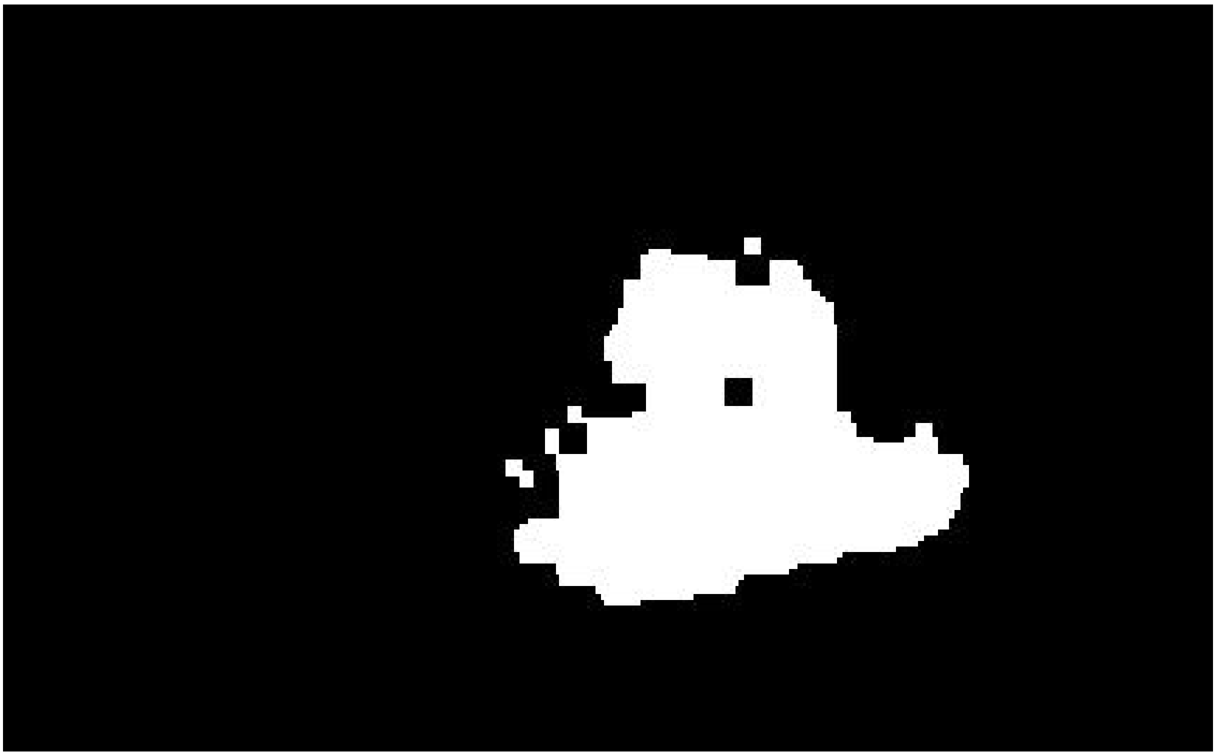

Figure 3.8: The mask after noise has been removed.

3.3 Shadow Removal

The Gaussian Mixture model used to segment the image is unable to distinguish pixels representing shadows from those of the vehicles and pedestrians we wish to track. It is desirable to remove these pixels to obtain just the groupings that correspond to objects of interest. The technique used in this project, involves comparing the hue and saturation values of each pixel in the foreground regions with their correspondences in an image representative of the background. The shadow removal function has two coloured images as inputs, one of the background and one of the current frame with all the background pixels values set to 0. These images are then converted to HSV in order to retrieve hue and saturation information. A pixel can be labelled as belonging to a shadow if its hue and saturation values differ from those of the related background pixel in a certain way. The output of the function is a binary mask with shadow pixels removed. An outline of the process can be seen in Figure 3.9.

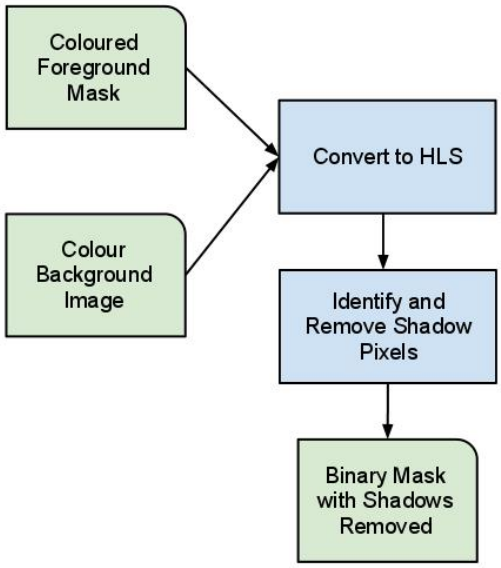

Figure 3.9: Shadow removal process.

It was necessary to generate a coloured background image for each frame as the Gaussian model only provides a foreground mask. This involved altering and rebuilding some functions from the native OpenCv library. Modification of the model was needed to generate RGB values for each background pixel. These values were calculated by selecting the mean value of the Gaussian curve with the greatest weight for that particular channel.

Each pixel in the current frame which is not already considered part of the background has its luminance and saturation values compared to those of the corresponding pixel in the background image. This method of discrimination relies on the assumption that differences in saturation between shadow and background follow a predictable pattern. A pixel is classified as shadow if it meets the following criteria:

- luminance is less then that of the corresponding background pixel
- luminance is greater than minimum allowed luminance

- saturation is less then that of the corresponding background pixel plus a small increase

The value for the minimum allowed luminance is calculated as a percentage of the background pixels luminance.

Figure 3.10 shows input to the shadow detection function on the left. On the right of the figure, a binary image of the foreground with pixels classified as shadow overlaid in orange. This technique worked quiet successfully and provided significant improvement to the segmentation process. However, the technique could not remove all shadow pixels. For regions very close to a car where luminance is particularly low, shadows could not be distinguished from wheels and the under-body of the car. It is also evident that some moving object pixels are mistakenly classified as shadow. Further Closing operations were used to fill in any holes that resulted in the outputted binary mask.

Figure 3.10: Shadow Removal

3.4 Finding Contours

Now that a segmented binary image has been created, it is required to group the connected foreground pixels into sets, each corresponding to an individual object in the scene. This can be accomplished by searching for contours in the binary image. A contour is simply a series of points representing a curve in an image. The contours in the binary image are the boundaries between positive and negative regions. The algorithm used by the native OpenCv functions to carry this process is described here[20]. Using this process, a set of contours is calculated from the binary mask in each frame with each contour representing a moving object in the captured scene.

Chapter 4

Object Tracking

The next step in process of the detection system is tracking. A contour in the current frame must be matched to a contour in the previous frame that describes the same moving object. Tracking moving objects from frame to frame allows the collection of information over time. For example, the direction of a car can be calculated and the car can be ignored by the system if it is travelling away from the traffic lights. This project made use of centroid tracking which is simpler but computationally cheaper then other methods. Centroid tracking relies on the assumption that the distance between the centroids of a contour in consecutive frames is always below a certain threshold.

4.1 Centroid Calculation

The centroid of a contour is calculated from the contours spatial moments. The moment of a contour can loosely be described as a summation of the values over all the contour's pixels. The formula for a spatial moment is given by Equation (4.1).

$$M_{m,n} = \sum x^m y^n P_{x,y} \tag{4.1}$$

where

- m and n are the x and y orders of the moments

- $P_{x,y}$ is the value of a pixel at position (x,y)

Using binary images, $P_{x,y} = 1$ for all points on the contour and $M_0 0$, the zero ordered moment now provides the area. This is also equivalent to the length, since the contour is made up of a series of boundary points. The x and y pixel positions of the centroid of a contour can be calculated from the zero and first order moments using Equations (4.2) and (4.3).

$$Centroid_x = M_{10}/M_{00} \tag{4.2}$$

$$Centroid_y = M_{01}/M_{00} \tag{4.3}$$

4.2 Track Assignment

Once the centroid of a contour has been calculated, we attempt to match that contour to a particular track. Each track accumulates information about a particular tracked object over time. The matching is based on the euclidean distance between the position value of the contour centroid and a value stored in the track. The tracks are updated with each new frame processed and even if a contour is not matched to a track in a particular frame then it can be matched again in the next frame. A new track is assigned to any contour which does not match any of the existing ones. Contours which have an area less than a minimum allowed size are disregarded as these are most likely caused by noise.

The following psuedocode illustrates the track assignment algorithm:

```
For each contour found in the current frame
    If(area > Minimum Allowed Area)
        Calculate Contour Centroid
        For each Track
            Calculate distance between contour and track centroids
        Find track with minimum distance
        If(minimum distance < Maximum Allowed distance)
            Assign contour to track with minimum distance
            Update track
        If(No match found)
            Assign new track
        End If
    End If
End For
```

In general the tracking worked quite well, vehicles and pedestrians could be followed from the point when they enter the scene to the point when they leave. The tracking can be visualised by assigning each track a distinct colour. A tracks colour is then used to draw a bounding rectangle of the contours matched to that track. The technique encountered difficulties in circumstances where vehicles or pedestrians overlapped in video. When this occurred the tracks were usually merged, with the smaller object becoming amalgamated into the track of the larger one.

It is also expected that this technique would encounter difficulties if moving objects become partially or completely occluded by elements of the background. This scenario did not occur in the sample data used for used for this project but it is expected that occlusions would cause large variance in the contours from frame to frame and distance between centroids may fall outside the permitted range.

Figure 4.1 shows the bounding rectangles of successfully tracked contours of vehicles and pedestrians and Figure 4.2 shows shows a scene a few seconds later when the tracks become merged.

Figure 4.1: Bounding Boxes of Tracked Objects

Figure 4.2: Merged Tracks

Chapter 5

Classification

Classification separates and labels the tracks of the moving objects into distinct classes. The vehicle detection system in this project only aimed to identify two classes of object, vehicles and pedestrians. The original intent of this project was to use a Bayesian classifier for this goal, however due to limits due time constraints and limited training data this proved unsuccessful. Instead a simpler shape based classification was used in order to prove the concept of the application.

5.1 Height to Width Ratio

The classifier used based on shape, in particular the height to width ratio of object contours. This classification is justified by the fact that contours of pedestrians show a greater height than width and vehicles the reverse. Tracks are only labelled if they maintain a certain level of persistence. The persistence is measured by the frequency of the tracks update, only tracks which have been updated in a large percentage of recent frames are updated. Tracks which have received no update after a fixed number of frames are removed entirely.

5.2 Calculating Direction

Only vehicles and people approaching the junction want to be considered when signalling the lights. The direction of travel of a tracked object can be used to determine whether or not it is moving towards or away from the camera. Objects moving away from the camera are not of concern when sequencing the traffic lights. A coarse calculation of a tracked objects direction in the vertical axis can be accomplished by calculating the difference in the y position of the tracked objects centroid over consecutive frames. If the difference value is positive then the object is moving towards the camera, otherwise the object can be ignored.

This quite basic classification was reasonably successful providing that moving objects in the scene did not overlap. Correct labellings of a vehicle and a pedestrian can be seen in Figure 5.1 and a group of pedestrians which were misclassified as a vehicle can be seen in Figure 5.2. Correct detection is not essential for system as the traffic lights can always revert to a default behaviour. If the system detects a sudden large variation in a particular tracks area then it can be aware that the classification is not reliable and fall back on the usual timed lighting sequence.

Figure 5.2: Two pedestrians classified.

Figure 5.1: Correctly labelled car and pedestrian

Figure 5.3: A busy scene with correctly labelled objects.

Figure 5.4: A group of pedestrians is mistakenly classified as a vehicle.

5.3 Naïve Bayes Classifier

A naïve Bayesian classifier would provide a more sophisticated probabilistic classification. Unfortunately this was not fully integrated into the system due to time constraints and a lack of sufficient training data. The classifier is first trained with labeled sample data to create the conditional probabilities for a number of object features. In the case of this project features such as height to width ratio, position in the scene and speed could be used to classify the detected objects.

Given a set of features from a track of an unknown class of object, we can determine the probability that the object belongs to a particular class by calculating the joint probability of the class and the set of features using the formula in Equation (5.1).

$$P(C, f_1, f_2, f_3, ...f_n) = P(C) \prod_{1 \leq k \leq n} P(f_k|C) \tag{5.1}$$

where f_i are features and C is the class

The class that maximises the joint probability is the one which is most likely to be present in the scene. By making the bold assumption that features are independent, the classifier provides a computationally less complex method of calculating joint probabilities. Naïve Bayes is a simple probabilistic classifier which is known to perform well when only small amounts of training data are available.

Chapter 6

Traffic Simulation

This chapter describes the design of a traffic simulator which was used to measure the impact that a vehicle detection system would have on journey times if installed at a traffic controlled intersection. It models a four-way signaled junction and approaching vehicles. The model allows for the simulation of vehicles as they pass through the junction and measures the times taken to clear a stretch of a road extending either side of the where the roads meet. Comparisons can then be made between a default, timed lighting sequence and light sequences triggered by the vehicle detection system. The simulator application was developed using Java and a graphical interface provides a visualisation of the simulated scenarios.

6.1 Design

Each simulation created by the application starts a timer which is used to periodically create vehicles and trigger the traffic light changes when the default sequencing operation is in effect. Figure 6.1 shows the main classes involved and some of the key interactions between them.

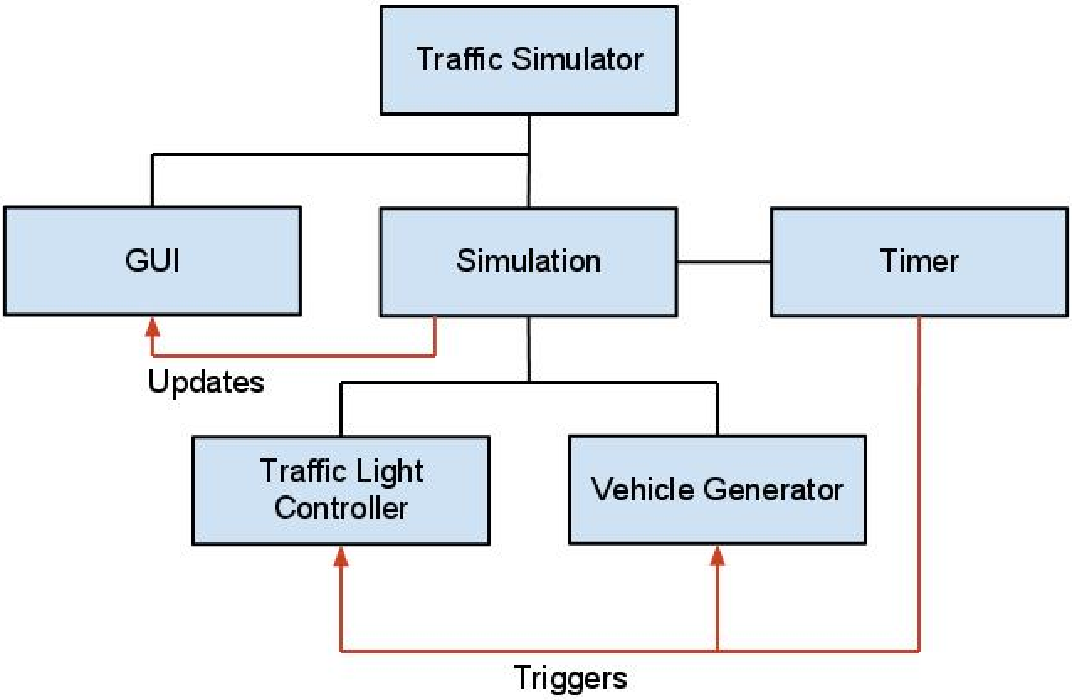

Figure 6.1: The main classes and key interactions of the Traffic Simulator.

6.2 Traffic Model

The following section describes the model used to simulate traffic at a signaled crossroads. The traffic lights used in the model have only two phases. One phase giving right of way only to vehicles approaching from the east and west and the second phase from the north and south. Vehicles are generated to approach periodically from random direction. The traffic flow can be adjusted by varying the interval between when the vehicles are created. Flow is defined as the number of cars passing a certain point per unit time. The flow entering the junction can be increased by decreasing the time interval between when new cars are generated.

The traffic lights can function under two sequencing operations.

- Default operation: Each phase of the traffic lights lasts for a fixed duration.
- Vehicle detection operation: Operates on a timer, the same as the default operation except when a vehicle is detected and the timer is preempted. If there are multiple detections on conflicting routes then no preemption occurs.

The model also makes the following assumptions:

- All vehicles are travelling at constant velocity. The model does not consider vehicles acceleration and deceleration when they must brake to stop at a red light. If these factors were taken into account, it would be expected that journey times would increase in situations where vehicles were forced to stop at red lights.
- All vehicles travel straight through the junction. Since vehicles only turn after they pass through the traffic lights, any impact on journey times would be unrelated to the light sequencing operation.

6.3 Graphical Interface

The graphical interface displays a scaled view of the simulation as well as a number of application controls. These allow a user to dynamically alter application parameters such as the vehicle speed, traffic light operation and the time interval between default traffic light changes. The graphical interface was useful for visualising, testing and demonstrating the simulator. A screenshot of the simulators interface is seen in Figure 6.2.

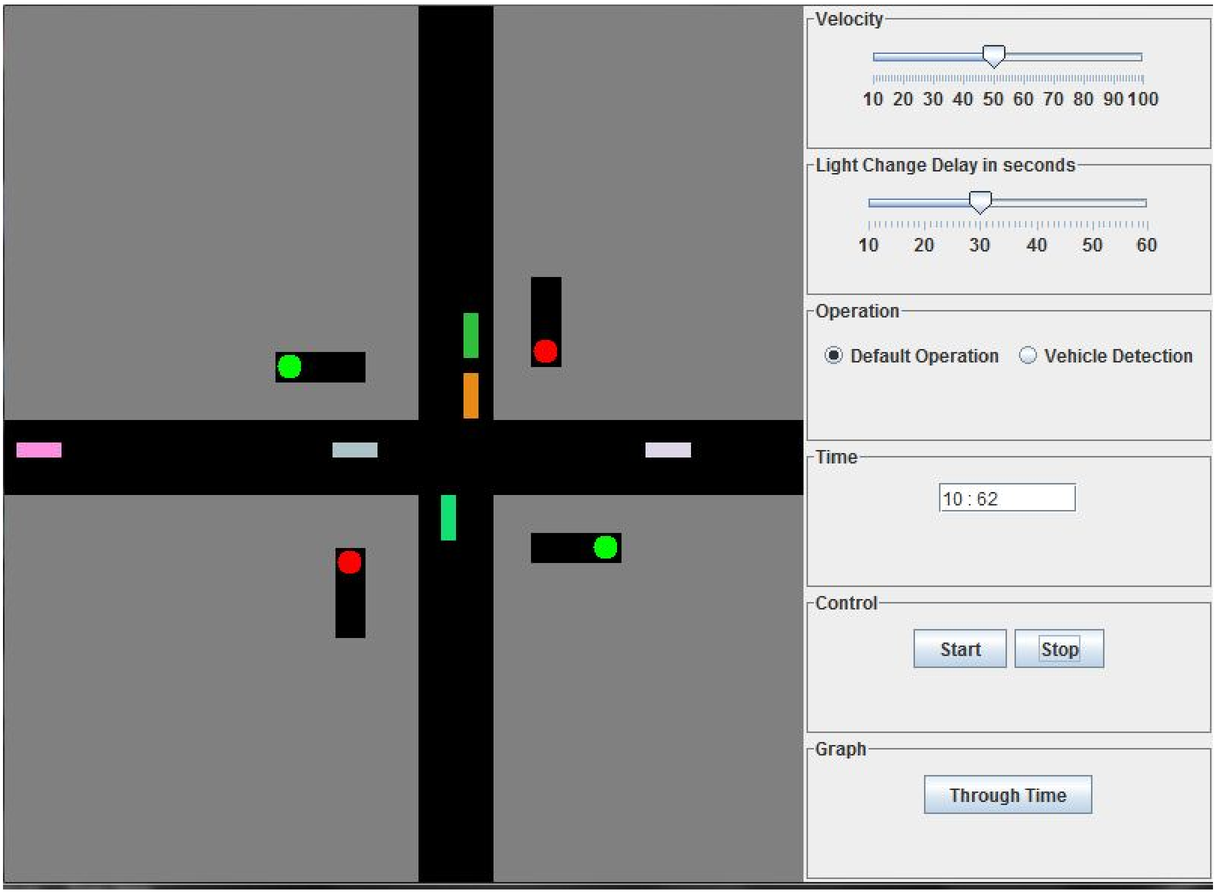

Figure 6.2: The graphical interface of the simulator.

6.4 Results

The simulation was run multiple times for various levels of traffic flow for each of the two traffic light sequence operations. Vehicles speed was set to 50 km/h and a detection distance of 25 metres was used. The time interval for triggering a change in the default operation was set to 30 seconds. The average through time was measured for each of the simulations and times compared for the different levels of traffic flow. The average through time is the average time taken for a vehicle to travel a certain distance spanning the junction.

The results showed, as was expected that for low levels of flow entering the junction, the detection system was able to reduce the average through time of vehicles. The graph in Figure 6.3 shows a plot of the average through time for a range of traffic flow levels. The average times are measured for both the default and detection modes of operation.

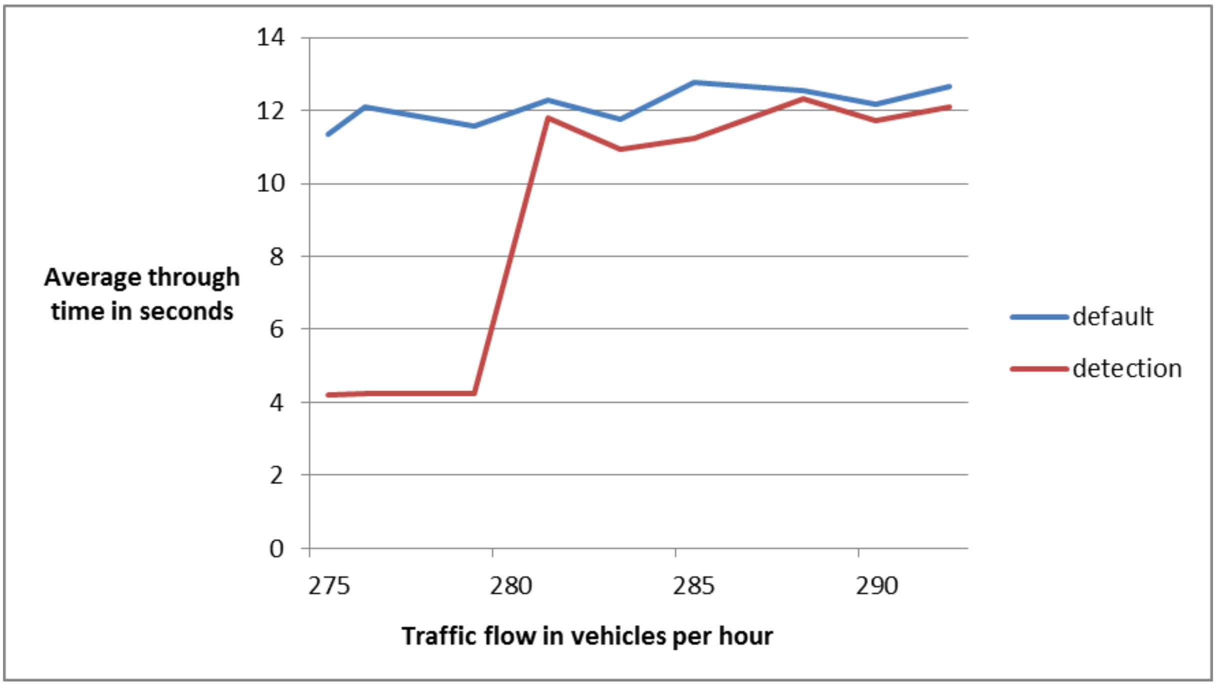

Figure 6.3: Comparison of average through-times of both modes of traffic light operation for increases in traffic flow.

It can be seen from the graph that as flow rises from 280 to 285 vehicles entering the junction per hour, the detection system becomes abruptly less effective at reducing journey times. At this point the detection system is overridden as vehicles are detected on multiple conflicting routes and the traffic lights revert to their default sequence. The varying traffic flow shows little effect on the default timed lighting operation.

Chapter 7

Conclusion

This project demonstrated that it is possible to use computer vision to detect and distinguish between vehicles and pedestrians by processing video from a camera monitoring a road scene. It was also shown by means of simulation that for low flows of traffic, such as those that might occur at quiet suburban junctions, a detection system could reduce journey times. As with many problems in computer vision, there are a multitude of techniques which could have been used to approach each of the steps involved in the traffic detection system. The difficulty is often in choosing the right techniques for the task at hand.

The mixture of Gaussians background model used proved very effective for segmentation. Moving object pixels in each frame were well separated from the background and could be grouped into connected regions. The downside was that model could not perform well enough to give real time execution even after optimising some of its input parameters for speed and reducing the number of frames processed per second.

The centroid based algorithm used was able to track moving objects from frame to frame. This worked well in the majority of cases but occasionally merged different objects when they overlapped or came too close together.

The Bayesian classifier intended to be used for classification was not successfully integrated into the system but it was shown that even a simple classification based on a height to width ratio of the moving objects was capable of distinguishing vehicles from pedestrians in most cases.

7.1 Limitations

There were some limitations to the detection system implemented in this project. Detection during adverse weather conditions and at night are two such scenarios which were not confronted in this work. Rain, snow or even heavy winds can have a detrimental effect on the segmentation process. Handling these scenarios would be important for a fully viable, real world system. Night detection would be particularly useful as roads are often quiet and the system would be beneficial at this time. Recognition at night and in low lighting conditions can require a very different process, where a vehicles most distinguishing features are its headlights. Also the classification in the system was limited to vehicles and pedestrians, cyclists and motorcyclists were not considered within the scope of this project.

7.2 Future Work

As well as addressing the limitations described in the previous section which were outside the scope of this dissertation, other possible future work could include using omnidirectional vision and integrating a probabilistic classification method.

Omnidirectional vision could be used to monitor all approaches to a junction with a single camera. Early in the project attempts were made to capture video using a camera pointed vertically towards a spherical mirror but there were difficulties obtaining an image with enough detail to process and getting the camera to focus correctly on the mirror. It would be interesting to see if a differently shaped mirror, perhaps parabolic would prove more successful.

Further progression of this project could also be made by integrating a functioning probabilistic classifier such as the Bayes classifier described in Chapter 5 of this work. The classifier could be trained with sufficient training data and performance could be examined.

Bibliography

[1] Clear 2006, classification of events, activities and relationships evaluation and workshop. http://isl.ira.uka.de/clear06/, June 2011.

[2] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 495 –501, June 1997.

[3] V. Bhuvaneshwar and P.B. Mirchandani. Real-time detection of crossing pedestrians for traffic-adaptive signal control. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 309 – 313, October 2004.

[4] N. Buch, J. Orwell, and S.A. Velastin. Urban road user detection and classification using 3d wire frame models. *Computer Vision, IET*, 4(2):105 –116, june 2010.

[5] A. Ghorayeb, A. Potelle, L. Devendeville, and E.-M. Mouaddib. Optimal omnidirectional sensor for urban traffic diagnosis in crossroads. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 597 –602, June 2010.

[6] S. Gupte, O. Masoud, R.F.K. Martin, and N.P. Papanikolopoulos. Detection and classification of vehicles. *Intelligent Transportation Systems, IEEE Transactions on*, 3(1):37 –47, mar 2002.

[7] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 696 – 701 vol. 1, June 2005.

[8] R. Khoshabeh, T. Gandhi, and M.M. Trivedi. Multi-camera based traffic flow characterization and classification. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 259 –264, October 2007.

[9] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 126 –131 vol.1, October 1994.

[10] P. Kumar, S. Ranganath, Huang Weimin, and K. Sengupta. Framework for real-time behavior interpretation from traffic video. *Intelligent Transportation Systems, IEEE Transactions on*, 6(1):43 – 53, mar 2005.

[11] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. In *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, pages 8 –14, oct 1998.

[12] Y. Liu and P. Payeur. Robust image-based detection of activity for traffic control. *Electrical and Computer Engineering, Canadian Journal of*, 28(2):63 –67, April 2003.

[13] D. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing*, 22(2):143–155, February 2004.

[14] Stefano Messelodi, Maria Modena, and Michele Zanin. A computer vision system for the detection and classification of vehicles at urban road intersections. *Pattern Anal. Appl.*, 8:17–31, September 2005.

[15] C. Papageorgiou and T. Poggio. Trainable pedestrian detection. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 4, pages 35 –39 vol.4, 1999.

[16] G. Scotti, A. Cuocolo, C. Coelho, and L. Marchesotti. A novel pedestrian classification algorithm for a high definition dual camera 360 degrees surveillance system. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III – 880–3, September 2005.

[17] G. Scotti, L. Marcenaro, C. Coelho, F. Selvaggi, and C.S. Regazzoni. A novel dual camera intelligent sensor for high definition 360 degrees surveillance. In *Intelligent Distributed Surveilliance Systems, IEE*, pages 26 – 30, feb 2004.

[18] C. Setchell and E.L. Dagless. Vision-based road-traffic monitoring sensor. *Vision, Image and Signal Processing, IEE Proceedings* -, 148(1):78 –84, February 2001.

[19] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 2 vol. (xxiii+637+663), 1999.

[20] S. Suzuki and K. Be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, April 1985.

[21] H. Veeraraghavan, O. Masoud, and N.P. Papanikolopoulos. Computer vision algorithms for intersection monitoring. *Intelligent Transportation Systems, IEEE Transactions on*, 4(2):78 – 89, June 2003.